X

The Network As Distributed Object Database*

Chien-Chung Shen
Department of Computer Science and Engineering
University of Texas at Arlington
Arlington, TX, 76019
cshen@cse.uta.edu

John Y. Wei
Bellcore
331 Newman Springs Road
Red Bank, NJ 07701
wei@bellcore.com

Abstract

As we move deeply into the information age, database has become an indispensable enabling technology and core component that supports enterprises' daily operations. Information ranging from employee and inventory data to customer profiles and marketing intelligence is stored inside databases where high-level queries may be applied to access (retrieve/update) more organized and intelligent information to perform mission-critical tasks.

In the area of telecommunications, it has been a well acknowledged fact that the network itself contains the definitive copy of information regarding itself. However, making the network information accessible as convenient and organized as contemporary databases has not been easy, due to the coexistence and non-interoperability of heterogeneous network elements and their associated management protocols. This defect may potentially jeopardize the operations consistency by creating a semantic gap of information between the management systems and the networks themselves.

In this paper, we describes a CORBA-based framework that helps to realize the notion of *the network as database*. This framework shall build abstraction layers that present a distributed object database abstraction for the networks to their managers. At the bottom layer, work of the X/Open & NMF Joint Inter-Domain Management is utilized and extended to encapsulate heterogeneous network resources with CORBA objects. On top of that, standardized network-level information models define *entity-relationship data models* which represent logical structures of the databases containing network resources. These distributed network resources, as represented by CORBA objects, collectively constitute the distributed object database which is organized according to the data model. Facilities, such as query, trigger, and view, are then provided to facilitate network management operations. We will illustrate how the framework may be realized via CORBA Object Services, and describe its application for modeling a WDM network.

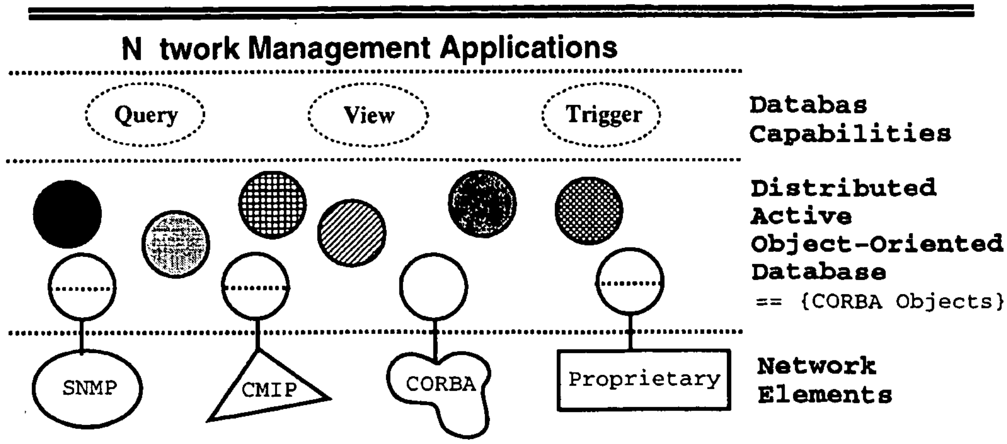
With the distributed CORBA object database representation of network resources, network management activities may be carried out as conveniently as database operations. The results not only facilitate effective and consistent network management operations, but also ease the integration with Service and Business Management Layers of the TMN architecture. For example, in the Service Management Layer, billing services may be supported by querying the network database on connection and utilization information. In the Business Management Layer, the notion of *data warehousing* may be realized, where *Online Analytical Processing (OLAP)* and *data mining* tools may access, analyze, and discover network information and knowledge, from the network database, to support business-oriented decisions such as network planning and re-engineering.

Keywords:

CORBA, Entity-Relationship Data Model, Network-Level Information Model, TMN, Joint Inter-Domain Management

*Prepared through collaborative participation in the Advanced Telecommunications Information Distribution Research Program (ATIRP) Consortium sponsored by the U.S. Army Research Laboratory under the Federated Laboratory Program, Cooperative Agreement DAAL01-96-2-0002.

The Network as Distributed Object Database



As we move deeply into the information age, database has become an indispensable enabling technology and core component that supports enterprises' daily operations. Information ranging from employee and inventory data to customer profiles and marketing intelligence is stored inside databases where high-level queries may be applied to access (retrieve/update) more organized and intelligent information to perform mission-critical tasks.

Typically, database design begins with defining a data model based on application-specific knowledge to provide an abstract description of the real world objects (entities and their associations). The data model represents the logical structure of the database and facilitates high-level queries to access organized information.

In the area of telecommunications, it has been a well acknowledged fact that the network itself contains the definitive copy of information regarding itself. However, making the network information accessible as convenient and organized as contemporary databases has not been easy, due to the coexistence and non-interoperability of heterogeneous network elements and their associated management protocols. This defect may potentially jeopardize the operations consistency by creating a semantic gap of information between the management systems and the networks themselves.

In this paper, we describes a CORBA-based framework that helps to realize the notion of *the network as database* [1]. This framework shall build abstraction layers that present a distributed object database abstraction for the networks to their managers. At the bottom layer, work of the X/Open & NMF Joint Inter-Domain Management is utilized and extended to encapsulate heterogeneous network resources with CORBA objects. On top of that, standardized network-level information models define *entity-relationship data models* which represent logical structures of the databases containing network resources. The distributed network resources, as represented by CORBA objects, collectively constitute the distributed object database which is organized according to the data model. Facilities, such as query, trigger, and view, are then provided to facilitate network management operations. The above figure depicts the concept pictorially. With the database abstraction, both network management functionality and information can be seamlessly integrated with other aspects of enterprise activities such as service and business management.

The remainder of the paper is organized as follows. We first describe network-level information models for transport networks. How the information models may be used to define entity-relationship data models is described next. We then describe how the proposed framework may be realized by making use of the work of Joint Inter-Domain Management and CORBA Object Services We also illustrate that the distributed object database representation of network information complys with the TMN logical layered architecture. Other database capabilities, such as trigger and view are described next to support timely reaction to network events and customer network management. We also demonstrate how the framework is used to model WDM networks. Finally, we conclude with integrating the framework with service and business management.

Network–Level Information Models

* Network–level information models
  - TINA–C: Network Resource Information Model
  - ITU SG 4: Common Information Viewpoint

* Represent a global view of transport networks and consider how individual network elements are interconnected and configured to provide end–to–end connectivity

* Notations
  - GDMO–like notation to define information objects
    - e.g. subnetwork, link, linkTP
  - Generic Relationship Model (GRM) to define how information objects are related
    - e.g. partitioning, linkBinds, contains

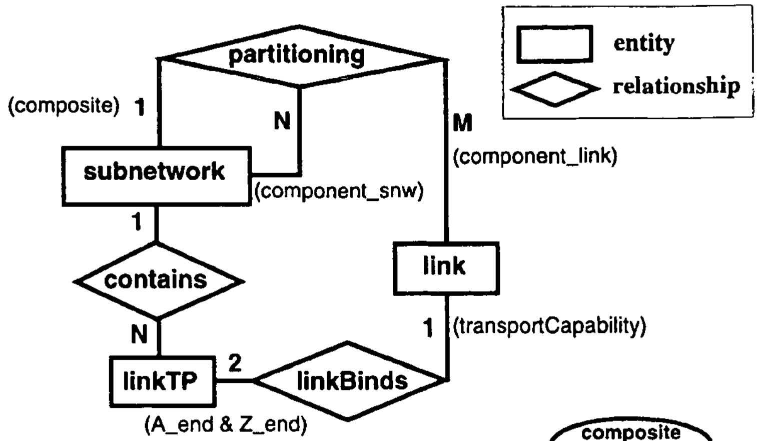
Recent activity on information modeling within standard body and consortium has been focusing on defining network-level information models. For example, ITU-T Study Group 4 had proposed the Recommendations G.853-01: Common Information Viewpoint [2] and TINA-C had defined the Network Resource Information Model Specification [3]. These network-level information models represent a global view of transport networks and consider how individual network elements are interconnected and configured to provide end-to-end connectivity. The information models contains definitions of information objects and their relationships that provide transmission and switching technology-independent information specifications of network resources. They describe transport networks abstractly in terms of network element, aggregation of network elements, topological relationship between the elements, connection end-point, and transport connection. In particular, both models have adopted a GDMO-like notation to define information objects and used GRM (General Relationship Model) to indicate how information objects are related. For example, the basic structure of transport networks could be modeled by the **subnetwork** information object which contains other subnetworks that are grouped together for topological reasons. Its object class is defined as follows.

```
subnetwork INFORMATION OBJECT CLASS
CHARACTERIZED BY
   DEFINED AS
      "A subnetwork information object represents a G.805:1995 subnetwork."
   ATTRIBUTES
      signalIdentification, subnetworkName;;
```

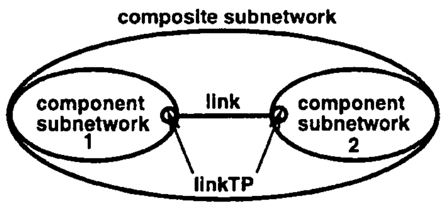
In addition, relationships are indicated by GRM templates, where each GRM template identifies *roles* in the relationship and identifies information objects that may play each of these roles. The **partitioning** relationship regarding the above defined information object is defined by the following GRM templates.

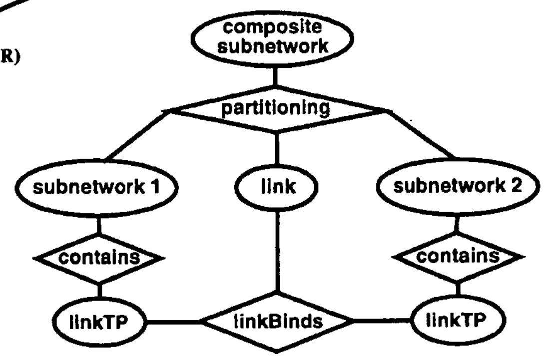
```
partitioning RELATIONSHIP CLASS
   DEFINED AS
      "The partitioning relationship class describes the relationship that exists
      between a composite subnetwork and component subnetwork and link instances that
      are part of its decomposition due to partitioning."
   ROLE composite
      COMPATIBLE-WITH subnetwork
      ROLE-CARDINALITY-CONSTRAINT (1..1)
   ROLE component_snw
      COMPATIBLE-WITH subnetwork
      ROLE-CARDINALITY-CONSTRAINT (1..N)
   ROLE component_link
      COMPATIBLE-WITH link
      ROLE-CARDINALITY-CONSTRAINT (1..M)
```

Information Model as Entity–Relationship Data Model



(ER)





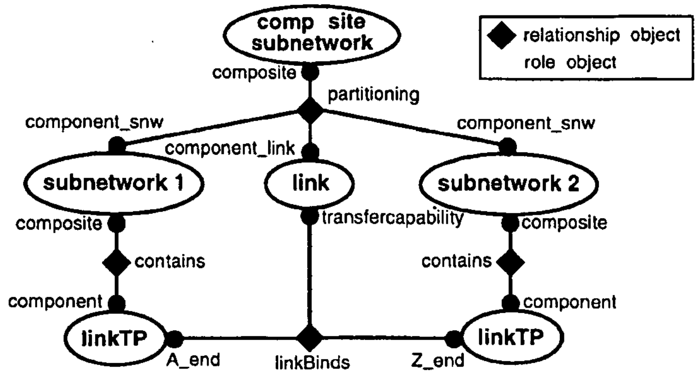
(A)

(B)

Given the defined network information objects and relationship classes, figure (ER) depicts a schematic diagram which defines an *entity-relationship data model* [4] to represent network-level information resources and their associations. The data model represents the logical structure of the database, and constraints what objects are legitimate in the database and how they are related. This figure may be interpreted as follows. A composite subnetwork may be partitioned into a set of component subnetworks and a set of links. Each link is bound to two link termination points. And link termination points are contained within their corresponding subnetworks, respectively.

Based on the entity-relationship data model, the transport network shown in figure (A) will have the corresponding database, shown in figure (B), representing its network-level information model. These objects, representing both entity and relationship, collectively constitute an distributed object database which is organized according to the data model.

CORBA-Based Realization



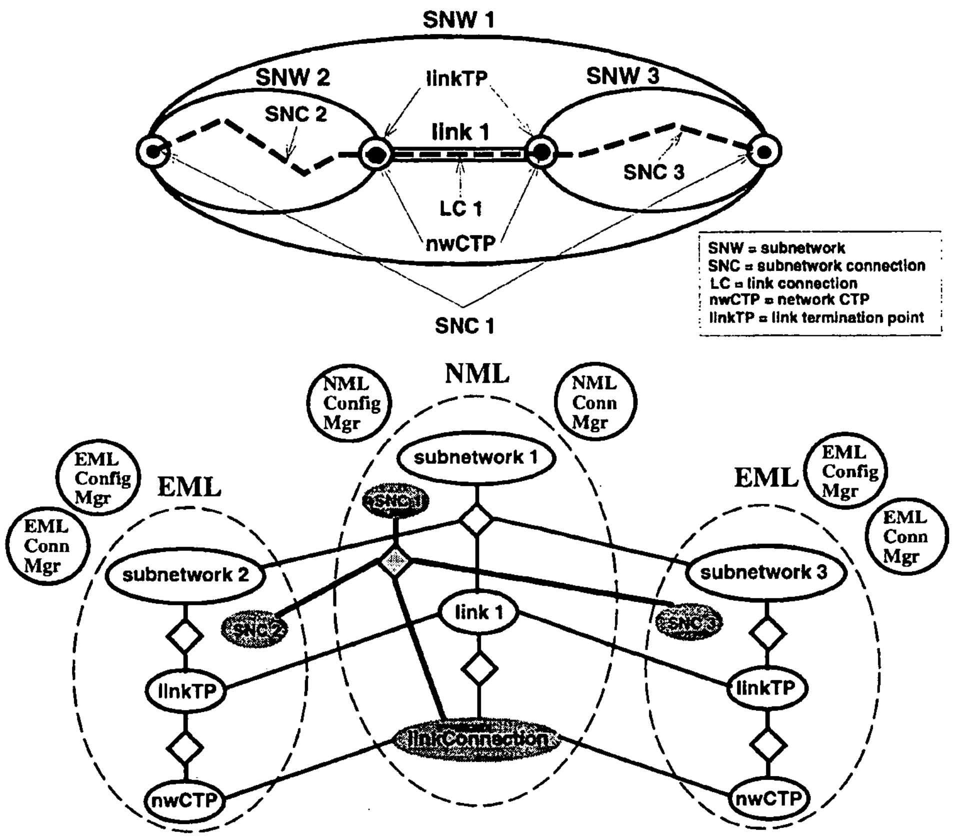
*** X/Open & NMF Joint Inter–Domain Management**

*** CORBA Object Services**
– Relationship, Lifecycle, Query, Transaction, Concurrency Control, Persistency, Event

Now, we describe how the 'network as distributed object database' framework may be realized using the emerging CORBA technology. In particular, we have taken advantage of the Joint Inter-Domain Management effort [7], and utilized CORBA Object Services.

Joint Inter-Domain Management (JIDM) The key idea of JIDM lies in the use of the CORBA technology to hide heterogeneity of network management protocols and to enable a distributed object computing environment. To hide management protocol heterogeneity, the CORBA IDL is used as the generic notation for specifying heterogeneous network resources, where CMIP GDMO and SNMP MIB specifications are translated into CORBA IDL specifications (**specification translation**). Gateway CORBA objects, that invoke CORBA on one side and either SNMP or CMIP on the other, translate interactions between domains to enable interworking between heterogeneous management systems (**interaction translation**). In our framework, we make use of JIDM to encapsulate heterogeneous network resources with common CORBA objects. These CORBA objects constitute a distributed object database to facilitate network management operations.

CORBA Object Services CORBA Object Services is a collection of interfaces and objects that support basic or near system-level functions for using and implementing objects. We have made use of the following Object Services in our framework: **Relationship Service** plays the central role in the framework by providing for creating, deleting, navigating, and managing relationships between objects. **Lifecycle Service** provides capability for creating and deleting distributed objects. For instance, the 'SNC' (subnetwork connection) objects may be created and deleted dynamically according to connection set-up and tear-down requests. **Query Service** provides query operations on collections of objects that may return (select), insert, update, and delete collections of objects. **Transaction Service** provides support to ensure that a computation involving more than one object provides ACID (Atomicity, Consistency, Isolation, and Durability) property. For instance, in the *provisioning*-style end-to-end connection set-up, network resources should be allocated and released *atomically* to ensure network consistency. The corresponding network resource objects involved should be created/deleted and manipulated in an *all or nothing* fashion via Transaction Service. **Concurrency Control Service** defines how an object mediates simultaneous access by more than one client such that it and objects it accesses remain consistent and coherent. It will provide the desired *isolation* guarantees to support Transaction Service. **Persistent Object Service** provides interfaces to the mechanisms used for retaining and managing the persistent state of objects in a database independent manner. It will provide the desired *durability* guarantees to support Transaction Service. **Event Notification Service** provides an *event channel* mechanism that decouples the communications between objects to allow multiple event *supplier* objects to communicate with multiple event *consumer* objects asynchronously. For example, a *link-cut* event may be communicated from the affected link object to the SNC object to notify its occurrence.



Based on the distributed object database representation of network resources, we now illustrate how the database structure complys with the TMN logical layered architecture [5].

To facilitate management interoperability, the TMN architecture defines an integrated management framework, where management tasks are partitioned into (logical) layers. Among them, the Network Management Layer (NML) controls and monitors the network view of all network elements within its domain, while the Element Management Layer (EML) controls and monitors a subset of the network elements. Although the current TMN standard makes use of the OSI management framework, it will not be exclusively tied to the X.700 OSI management methods. For instance, the effort of Reference Model of Open Distributed Processing (RM-ODP) [6] encourages TMN to use a *distributed object computing* paradigm, in addition to the *manager/agent* paradigm.

For example, the above figure presents a more elaborated subnetwork architecture together with a subnetwork connection. The corresponding network-level information model is represented as a distributed object database shown. These database objects are organized into NML and EML layers based on subnetwork partition, and they can be encapsulated within separate processes and located within different management stations. In addition, NML and EML layer management objects (configuration and connection) are responsible for stewarding the corresponding objects to provide management functionalities. For instance, according to network operator's connection set-up and tear-down requests, the NML connection manager may create and delete the NML-layer SNC and linkConnection Objects within the database, and delegate two EML connection managers to create and delete the corresponding EML-layer SNC objects.

Other Database Capabilities: Trigger and View

*** Trigger for Active Object Database**
- react to network events in a timely manner

*** View for Customer Network Management**
- Abstraction Mechanism
- Security Mechanism

In addition to the basic database capabilities (such as Query and Transaction) provided by CORBA Object Services, others are needed to facilitate more versatile network management operations. Among them, trigger for active object database and view for customer network management are of particular importance. Their concise descriptions are given below, and work is in progress to integrate them with the CORBA architecture.

Trigger for Active Object Database

The capability of *active database* has been proposed to satisfy the requirements of applications that must react to events in a timely manner. Network management is a typical application that must react to the network faults by isolating and correcting them in a timely manner so as not to jeopardize network traffic and services.

To provide the capability, relational database systems usually equip a table with triggers which are invoked automatically whenever the table is the target of an insert, delete, or update operation. They are provided for enforcing database-specific rules such as referential integrity constraints.

In general, active objects need to possess the capability for specifying their *active behavior* in the form of *Event-Condition-Action* rules, where an event determines when the rule should be triggered, the condition states whether the action should be executed, and the action determines how the object should respond [8]. In our framework, the goal is to make CORBA objects active.

View for Customer Network Management

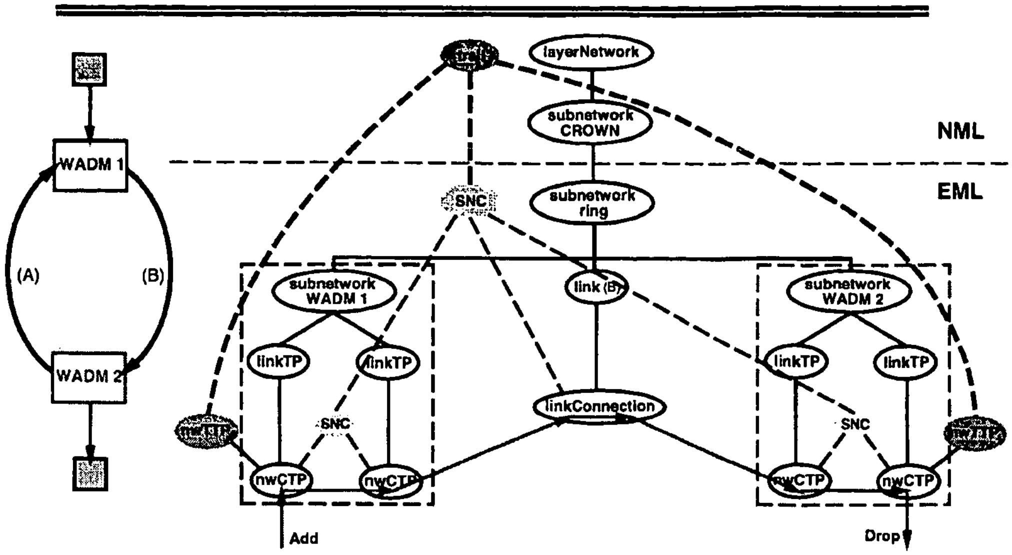
Customer Network Management (CNM) describes a set of capabilities provided by public network providers that enable the customers to monitor and control parts of the *subscribed* network infrastructure and services. Within our framework, CNM may be provided with customer-specific *views* on the public network which contain network management information that is only relevant to the specific customer.

In relational database, a view is a *virtual* table constructed by forming the table from fields in one or more related base tables by some selection criteria. A view presents a current, but restricted, 'window' into the database, and is always up to date with the latest changes made in any of the base tables used to compose it. The view mechanism has the following two key features which form the basis for developing customer network management.

- Abstraction Mechanism: Views provide a powerful abstraction mechanism that allows users to focus on just the data that is of interest to them and to ignore the rest.
- Security Mechanism: By hiding irrelevant information from a user, views restrict user from accessing other information in the base tables, which provides an efficient access control mechanism.

In our framework, the goal is to extend CORBA objects with view capability.

WDM Network Information Model

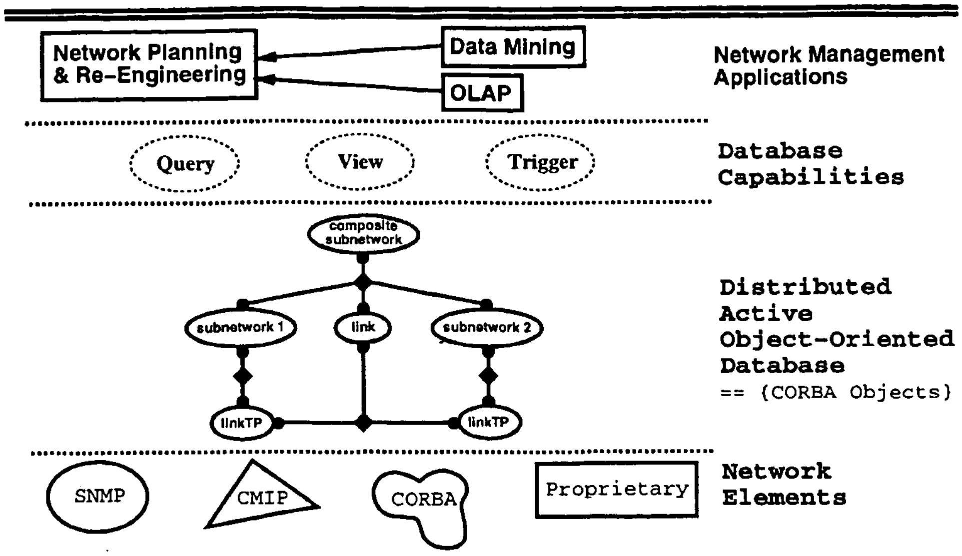


We now apply the described framework to model a WDM (Wavelength-Division-Multiplexing) network. NML and EML configuration managers collaborately maintain a network-level information model of distributed CORBA objects, which consists of the following CORBA objects.

- **Layer Network** is a transport network carrying a particular characteristic information.
- **Subnetwork** contains other subnetworks and links that are grouped together for topological or other reasons. The smallest subnetwork is a physical NE.
- **Link** represents physical connectivity (fiber) between two subnetworks.
- **Link Termination Point (LinkTP)** represents an end point of a link, and contains a number of nwCTPs depending on the number of wavelengths supported in the fiber.
- **Network Connection Termination Point (nwCTP)** represents an end point of a subnetwork connection or link connection.
- **Network Trail Termination Point (nwTTP)** delimits a layer network. These are the access points for client layer network elements.
- **Subnetwork Connection (SNC)** is the resource that transfers information between two nwCTPs of the subnetwork.
- **Link Connection** is the resource that transfers the characteristic information between two subnetworks.

Given the two-WADM (Add-Drop Multiplexer) ring network, the corresponding network-level information model is constructed as follows. A layerNetwork object consists of a CROWN subnetwork object which in terms consists of a ring subnetwork object. The ring subnetwork object consists of subnetwork objects WADM 1 and 2, which are inter-connected by link objects A and B. WADM 1 has three linkTP objects denoting three connecting fibers. We show only two of them: one denoting the incoming access link, and the other denoting the outgoing link B. The linkTP of WADM 1 associated with link B contains eight nwCTP objects representing eight different wavelengths supported on the fiber. We show a single nwCTP to denote an end point of a particular link connection (wavelength connection) which is contained by the link B object. The provisioning of an optical path is denoted by creating appropriate SNC objects which associate nwCTP and linkConnection objects.

The Network as Distributed Object Database



Motivated by the importance and effectiveness of database, this paper proposed the 'network as distributed object database' framework to present a unified management view of diverse management protocols and networking technologies. The framework, based on CORBA, builds abstraction layers that present a distributed object database abstraction for the networks to their managers.

With the distributed object database representation of network resources, network management activities may be carried out as conveniently as database operations. The results not only facilitate effective and consistent network management operations, but also ease the integration with Service and Business Management Layers of the TMN architecture. For example, in the Service Management Layer, billing services may be supported by querying the network database on connection and utilization information. In the Business Management Layer, the notion of *data warehousing* may be realized, where *Online Analytical Processing (OLAP)* and *data mining* tools may access, analyze, and discover network information and knowledge, from the network database, to support business-oriented decisions such as network planning and re-engineering.

References

[1] L. Bernstein and C.M. Yuhas, "Managing Telecommunications Networks," IEEE Network, November 1993.

[2] ITU-T, "Common Elements of the Information Viewpoint for the Management of a Transport Network," Study Group 4 Draft Recommendation G.853-01, June 1996.

[3] TINA-C, "Network Resource Information Model Specification," TINA-C Baseline Draft, May 1996.

[4] P. Chen, "The Entity-Relationship Model: Toward a Unified View of Data," ACM Transactions on Database Systems, Vol. 1, No. 1, March 1976.

[5] ITU-T, "Principles for a Telecommunications Management Network," Recommendation M.3010, 1993.

[6] ITU-T, "Reference Model for Open Distributed Processing," Recommendation X.901, 1993.

[7] X/Open & NMF, "Inter-Domain Management: Specification Translation," June 1996.

[8] A. P. Buchmann, "Active Object Systems," NATO ASI OODBS. pp. 201–224, 1993.

The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government.